

Predictive analytics in Smart education - Educational analytics

IDENTIFYING INFLUENTIAL LEARNING OBSTACLES OF STUDENTS



TEAM MEMBER DETAILS

- | | | |
|------------------------|----------------|----------------------------------------------------------------------------|
| 1. SUJAN BOSE B | (917721S034) | sujan@student.tce.edu |
| 2. VISHWAJITH J | (917721S040) | vishwajith@student.tce.edu |
| 3. SHARAN B | (917721S029) | sharanb@student.tce.edu |
-

Objective :-

Identifying the obstacles that hinder a student's learning progress is a crucial step towards improving the overall learning experience. It is an issue that has gained significant attention in recent years as educators seek to provide a more personalized and effective learning environment for students. In this context, “ **THE OBJECTIVE OF THIS STUDY IS TO IDENTIFY THE MOST INFLUENTIAL LEARNING OBSTACLES** and use **MACHINE LEARNING ALGORITHMS** to predict student performance based on these obstacles ”.

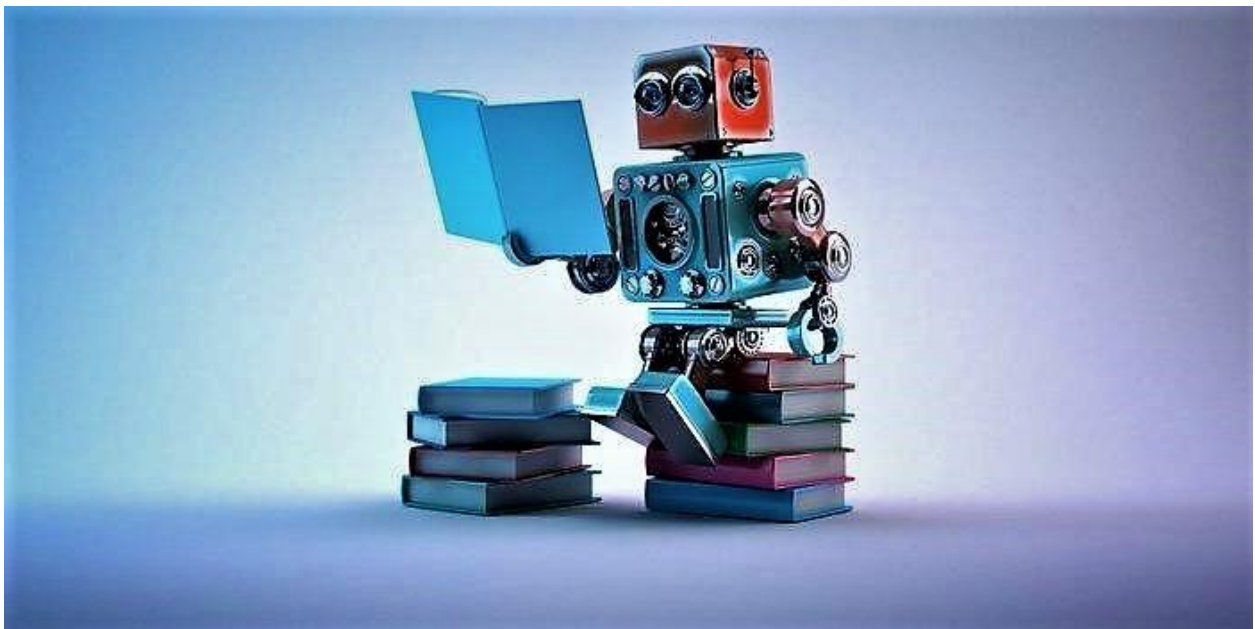
The first step in this study involves the identification of the learning obstacles that students face. These obstacles can be categorized into **academic** and **non-academic factors** that may impact a student's learning experience. **Academic obstacles** may include difficulties in understanding subject matter, inadequate resources, and lack of academic support. **Non-academic obstacles** may include factors such as socio-economic status, family dynamics, and health issues.

To identify these obstacles, educators can use various tools and techniques, such as **student assessments, surveys, and observations**. These methods can provide insight into the challenges that students face and allow educators to develop interventions that address these challenges. Additionally, data analysis tools, such as machine learning algorithms, can be used to **identify patterns** in student performance data and **predict potential obstacles that students may face**.

In this study, we focus on using machine learning algorithms to identify the most influential learning obstacles and predict student performance based on these obstacles. We begin by gathering data on student performance and the various academic and non-academic obstacles they face. This data is then used to **train** machine learning algorithms that can identify the most influential obstacles and **predict** student performance based on these obstacles.

Once the most influential learning obstacles have been identified, educators can develop interventions that address these obstacles. This may involve providing **additional support** and **resources**, **adapting teaching strategies**, and **providing targeted instruction to address specific academic or non-academic challenges**. The ultimate goal is to create a learning environment that is **personalized and effective**, addressing the specific challenges that each student faces and ensuring that they can achieve their full potential.

.



Experiment conducted :-

What data do you collect?

The data collected for the identification of learning obstacles can consist of various types of information, including numerical and categorical data. The values for each parameter can be either nominal or ordinal, depending on the nature of the data.

For instance,

The parameters used for prediction in this study are as follows:

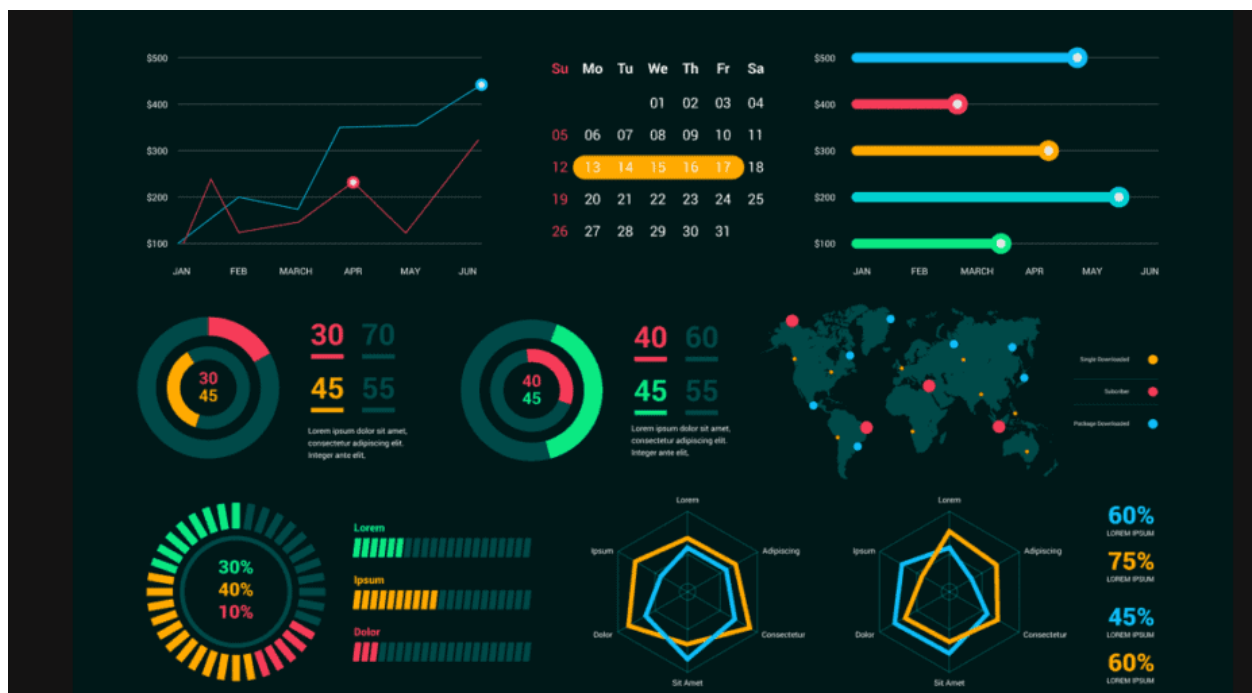
The following parameters are used for prediction:

- **Extensive Course Load:** This is a categorical data with four values, namely Low, Moderate, High and Very High, representing the extent of a student's course load.
- **Accommodation:** This is a categorical data with two values, namely Days Scholar and Hosteller, representing whether a student stays in college hostel or commutes from home.
- **Travel Time:** This is a categorical data with five values, namely Less than 30 minutes, 30-60 minutes, 1-2 hours, 2-3 hours, and more than 3 hours, representing the time taken by a student to travel from home to college.
- **Course Materials:** This is a categorical data with four values, namely Poor, Fair, Good, and Excellent, representing the quality of course materials provided to a student.
- **Health Issues:** This is a nominal data with two values, Yes and No, representing whether a student has health issues or not.
- **Financial Difficulties:** This is a nominal data with two values, Yes and No, representing whether a student is facing financial difficulties or not.
- **Anxiety:** This is an ordinal data with four values, namely Never, Rarely, Sometimes, and Almost all the time, representing the level of anxiety a student is experiencing.

- **Entertainment Time:** This is an ordinal data with four values, namely Below 1 hour, 1-2 hours, 2-4 hours, and Above 4 hours, representing the amount of time a student spends on entertainment activities.

- **Language:** This is a nominal data with two values, Yes and No, representing whether a student is comfortable with the language of instruction.

The values provided for each parameter are mapped to **numerical values** . This data can be used to build a predictive model to predict a student's **"SGPA"** and **"IDENTIFY THE MOST INFLUENTIAL LEARNING OBSTACLES"** based on the various learning obstacles they face.



Participants from whom you collect data?


The data for this study was collected through Google Forms. **Participants were students of TCE** who voluntarily completed the survey. The data collected was real-time and cross-sectional in nature, meaning it was collected at a single point in time from a sample of participants. The data collected includes **categorical, nominal, ordinal, and numerical variables**. This data collection approach is widely accepted in research studies, and the model developed using this data will be applicable to real-world scenarios, facilitating **decision-making based on real-time data and parameters**.



DATA COLLECTION MECHANISM ?

GOOGLE FORMS

URL :- <https://forms.gle/YzT8mtMWC3ZBrduP6>



Section 1 of 3

IDENTIFYING LEARNING OBSTACLES

Dear Participant,

We are conducting a research study to identify the **learning obstacles** faced by students while learning, and we need your help. Your participation in this study will greatly contribute to understanding the challenges students face in the learning process.

The survey consists of a series of questions asking about your learning experience in a specific research paper. Please read the paper carefully before answering the questions. Your responses will be anonymous and confidential.

Thank you for your time and participation.

Did the recent exam you completed have an **EXTENSIVE COURSE LOAD** (Internals , Semester) ? *

☐ Very High

☐ High

☐ Moderate

☐ Low

after section 1 Continue to next section

Section 2 of 3

days scholar

Description (optional)

What is your typical **TRAVEL TIME TO COLLEGE** on weekdays using public transport or personal vehicle? *

☐ less than 10 min

☐ 10-20 min

☐ 20-30

☐ 30-40

☐ more than 40 min

Section 3 of 3

general

Description (optional)

How easy it is for you to access **COURSE MATERIALS** (such as Labs , Apparatus , Online Resources , Technologies , Infrastructures) *

☐ Convenient

☐ Available

☐ Challenging

☐ Inadequate

Did you **EXPERIENCE** any **HEALTH ISSUES** during the course period that impacted your academic performance? *

☐ Yes

☐ No

Do you face any challenges understanding specific concepts or topics in your course due to **language barriers** ?

☐ Yes

☐ No

Are You a **Days Scholar** or **Hosteller**

☐ Days Scholar

☐ Hosteller

What is your typical **FINANCIAL DIFFICULTIES** or the Working in a Part-Time job for paying your Institution's Fees ? *

☐ Yes

☐ No

How often do you **EXPERIENCE** physical symptoms of **ANXIETY** (such as Exam Stress, sweating , Rapid Heartbeat , or Shivering)? *

☐ Never

☐ Occasionally

☐ Option 3

☐ Almost all the time

What's your average **ENTERTAINMENT TIME** (Movies , Social Media , Video Games) ? *

☐ Below 1 hour

☐ Between 1- 2 hrs

☐ Between 2 -4 hrs

☐ Above 4 hrs

please enter your **SGPA** score of last semester *

Short answer text

DATA PREPROCESSING :-

Import necessary libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Read the CSV file using pandas

```
df = pd.read_csv('SF_FINAL2.csv')
```

Separate the independent variables and dependent variable

```
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
```

Print the independent and dependent variables

```
print(X)
print(y)
```

Fill the missing values in the 'TRAVEL_TIME' column

```
df['TRAVEL_TIME'].fillna(value='less than 10 min', inplace=True)
print(df)
```

Replace the misspelled word 'Availabile' with 'Available' in the 'COURSE_MATERIALS' column

```
df['COURSE_MATERIALS'] = df['COURSE_MATERIALS'].replace('Availabile', 'Available')
```

Replace the similar values in the 'ENTERTAINMENT_TIME' column

```
df['ENTERTAINMENT_TIME'] = df['ENTERTAINMENT_TIME'].replace({  
    'Below 2 hrs': 'Between 1-2 hrs',  
    'Between 1 - 2 hrs': 'Between 1-2 hrs',  
    'Between 2 - 4 hrs': 'Between 2-4 hrs',  
    'Between 1-2 hrs': 'Between 1-2 hrs',  
    'Between 2-4 hrs': 'Between 2-4 hrs'  
})
```

Replace the values with correct spacing in the 'ENTERTAINMENT_TIME' column

```
df['ENTERTAINMENT_TIME'] = df['ENTERTAINMENT_TIME'].replace({  
    'Between 1-2 hrs': 'Between 1- 2 hrs',  
    'Between 2-4 hrs': 'Between 2 -4 hrs',  
})
```

Replace the values with correct format in the 'TRAVEL_TIME' column

```
df['TRAVEL_TIME'] = df['TRAVEL_TIME'].replace({  
    '30-40 min': '30-40',  
    '20-30 min': '20-30',  
    'Less than 10 min': 'less than 10 min',  
    'More than 40 min': 'more than 40 min'  
})
```

Replace the misspelled word 'Occasionly' with 'Occasionally' in the 'ANXIETY' column

```
df['ANXIETY'] = df['ANXIETY'].replace({  
    'Occasionly': 'Occasionally'  
})
```

save the modified dataframe to a csv file

```
df.to_csv("SF_FINAL2.csv", index=False)
```

Visualizing Categorical Data Using Pie Charts for Survey Responses in Python :-

```
import matplotlib.pyplot as plt
```

read in the csv file

```
df = pd.read_csv('SF_FINAL2.csv')
```

create a dictionary to store the column names and their values

```
col_dict = {'EXTENSIVE_COURSE_LOAD': 'Extensive Course Load',  
            'ACCOMDATION': 'Accommodation',  
            'TRAVEL_TIME': 'Travel Time',  
            'COURSE_MATERIALS': 'Course Materials',  
            'HEALTH_ISSUES ': 'Health Issues',  
            'FINANCIAL_DIFFICULTIES': 'Financial Difficulties',  
            'ANXIETY': 'Anxiety',  
            ' ENTERTAINMENT_TIME': 'Entertainment Time',  
            'LANGUAGE': 'Language'}
```

create a figure with multiple subplots

```
fig, axs = plt.subplots(3, 3, figsize=(15, 15))
```

loop through the columns and plot a pie chart for each one

```
for i, col in enumerate(col_dict.keys()):
```

```
    # calculate the value counts for the column
```

```
    counts = df[col].value_counts()
```

plot a pie chart in the appropriate subplot

```
axs[i//3, i%3].pie(counts.values, labels=counts.index, autopct='%1.1f%%')
```

```
axs[i//3, i%3].set_title(col_dict[col])
```

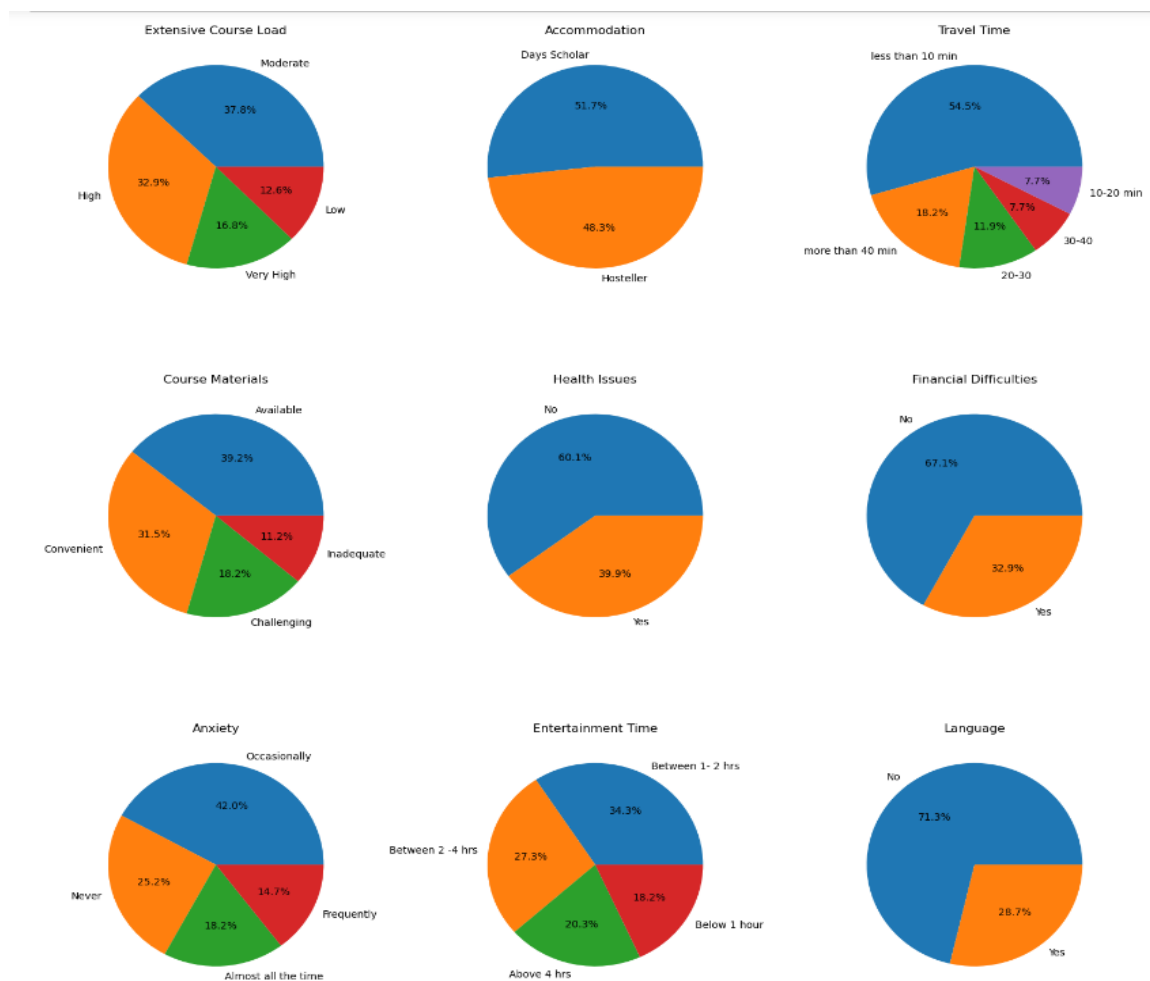
adjust the spacing between subplots

```
plt.tight_layout()
```

show the plot

```
plt.show()
```

OUTPUT :-



Encoding Categorical Data in a DataFrame

Load the data

```
data = pd.read_csv('SF_FINAL2.csv')
```

Define the columns to encode

```
columns_to_encode = ['EXTENSIVE_COURSE_LOAD', 'ACCOMDATION', 'TRAVEL_TIME',  
'COURSE_MATERIALS', 'HEALTH', 'FINANCIAL_DIFFICULTIES', 'ANXIETY',  
'ENTERTAINMENT', 'LANGUAGE']
```

Define the mappings for each column

```
extensive_course_load_mapping = {'Very High': 4, 'High': 3, 'Moderate': 2, 'Low': 1}
```

```
accomodation_mapping = {'Days Scholar': 0, 'Hosteller': 1}
```

```
travel_time_mapping = {'less than 10 min': 1, '10-20 min': 2, '20-30': 3, '30-40': 4, 'more than  
40 min': 5}
```

```
course_materials_mapping = {'Convenient': 4, 'Available': 3, 'Challenging': 2, 'Inadequate': 1}
```

```
health_mapping = {'Yes': 1, 'No': 0}
```

```
financial_difficulties_mapping = {'Yes': 1, 'No': 0}
```

```
anxiety_mapping = {'Never': 1, 'Occasionally': 2, 'Frequently': 3, 'Almost all the time': 4}
```

```
entertainment_time_mapping = {'Below 1 hour': 1, 'Between 1- 2 hrs': 2, 'Between 2 -4 hrs':  
3, 'Above 4 hrs': 4}
```

```
language_time_mapping = {'Yes': 1, 'No': 0}
```

Loop through the columns and encode each one

```
for column in columns_to_encode:
```

```
    if column == 'EXTENSIVE_COURSE_LOAD':
```

```
        data[column] = data[column].map(extensive_course_load_mapping)
```

```
    elif column == 'ACCOMDATION':
```

```
        data[column] = data[column].map(accomodation_mapping)
```

```
elif column == 'TRAVEL_TIME':
    data[column] = data[column].map(travel_time_mapping)
elif column == 'COURSE_MATERIALS':
    data[column] = data[column].map(course_materials_mapping)
elif column == 'HEALTH':
    data[column] = data[column].map(health_mapping)
elif column == 'FINANCIAL_DIFFICULTIES':
    data[column] = data[column].map(financial_difficulties_mapping)
elif column == 'ANXIETY':
    data[column] = data[column].map(anxiety_mapping)
elif column == 'ENTERTAINMENT':
    data[column] = data[column].map(entertainment_time_mapping)
elif column == 'LANGUAGE':
    data[column] = data[column].map(language_time_mapping)

# Save the encoded data to a new CSV file

data.to_csv('SF_FINAL2.csv', index=False)

# Print the first 10 rows of the encoded data

print(data.head(10))
```

Multicollinearity check using Variance Inflation Factor (VIF) :- FEATURE SELECTION

```
import pandas as pd

from statsmodels.stats.outliers_influence import variance_inflation_factor

# Load the dataset
data = pd.read_csv('SF_FINAL2.csv')

# Select the independent variables
X = data.drop(['SGPA'], axis=1)
y = data['SGPA'].values

# Calculate VIF scores for each independent variable
vif = pd.DataFrame()

vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

vif["features"] = X.columns

# Print the VIF scores
print(vif)
```

OUTPUT :-

	VIF Factor	features
0	7.255948	EXTENSIVE_COURSE_LOAD
1	3.044551	ACCOMDATION
2	5.293555	TRAVEL_TIME
3	7.727842	COURSE_MATERIALS
4	2.020994	HEALTH
5	1.698408	FINANCIAL_DIFFICULTIES
6	6.267665	ANXIETY
7	6.595051	ENTERTAINMENT
8	1.651851	LANGUAGE

Algorithms Applied On Data With Python Code :-

Decision Tree Regressor and **Random Forest Regressor** are two algorithms used to identify the most influential factor on SGPA. Decision Tree Regressor recursively splits data based on the feature with maximum information gain, while Random Forest Regressor creates a set of decision trees and aggregates their output. Both algorithms compute feature importance by calculating the percentage of samples that were split based on a particular feature.

CODES :-

#Decision Tree Regressor

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor

data = pd.read_csv('SF_FINAL2.csv')
X = data.drop(['SGPA'], axis=1)
y = data['SGPA'].values

# Create a decision tree regressor
dt = DecisionTreeRegressor(random_state=42)

# Train the model and compute the feature importance
dt.fit(X, y)
importance = dt.feature_importances_

# Print the feature importance
for i, v in enumerate(importance):
    print('Feature %0d: %.5f' % (i, v))

# Find the most important feature
most_important_feature_index = np.argmax(importance)
```

```
most_important_feature_name = X.columns[most_important_feature_index]
print('The MOST INFLUENTIAL PARAMETER is:', most_important_feature_name)
```

EVALUATION METRICS

```
from sklearn.metrics import mean_squared_error, r2_score
```

Make predictions on the training set

```
y_pred = dt.predict(X)
```

Calculate MSE and R^2

```
mse = mean_squared_error(y, y_pred)
```

```
r2 = r2_score(y, y_pred)
```

Print the results

```
print('MSE:', mse)
```

```
print('R^2:', r2)
```

OUTPUT :-

```
Feature 0: 0.14822
Feature 1: 0.03233
Feature 2: 0.06317
Feature 3: 0.25945
Feature 4: 0.11629
Feature 5: 0.10510
Feature 6: 0.06348
Feature 7: 0.19214
Feature 8: 0.01983
The MOST INFLUENTIAL PARAMETER is: COURSE_MATERIALS
```

```
MSE: 0.038983578431372554
```

```
R^2: 0.9556534241921991
```

#Random Forest Regressor

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor

# Load the dataset
data = pd.read_csv('SF_FINAL2.csv')

# Split the dataset into X and y
X = data.drop(['SGPA'], axis=1)
y = data['SGPA']

# Create a random forest regressor
rf = RandomForestRegressor(random_state=42)

# Train the model and compute the feature importance
rf.fit(X, y)
importances = rf.feature_importances_

# Print the feature importance
for i, v in enumerate(importances):
    print('Feature %0d: %.5f' % (i, v))

# Find the most important feature
most_important_feature_index = np.argmax(importances)
most_important_feature_name = X.columns[most_important_feature_index]
print('The MOST INFLUENTIAL PARAMETER is:', most_important_feature_name)
```

EVALUATION METRICS (Random Forest Regressor)

```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
```

Make predictions on the test set

```
y_pred = rf.predict(X)
```

Evaluate the model performance

```
r2 = r2_score(y, y_pred)
```

```
mse = mean_squared_error(y, y_pred)
```

```
mae = mean_absolute_error(y, y_pred)
```

Print the results

```
print('R2 score: ', r2)
```

```
print('MSE: ', mse)
```

```
print('MAE: ', mae)
```

OUTPUT :-

```
Feature 0: 0.13784
Feature 1: 0.08075
Feature 2: 0.11641
Feature 3: 0.20707
Feature 4: 0.06292
Feature 5: 0.06152
Feature 6: 0.11768
Feature 7: 0.17179
Feature 8: 0.04402
The MOST INFLUENTIAL PARAMETER is: COURSE_MATERIALS
```

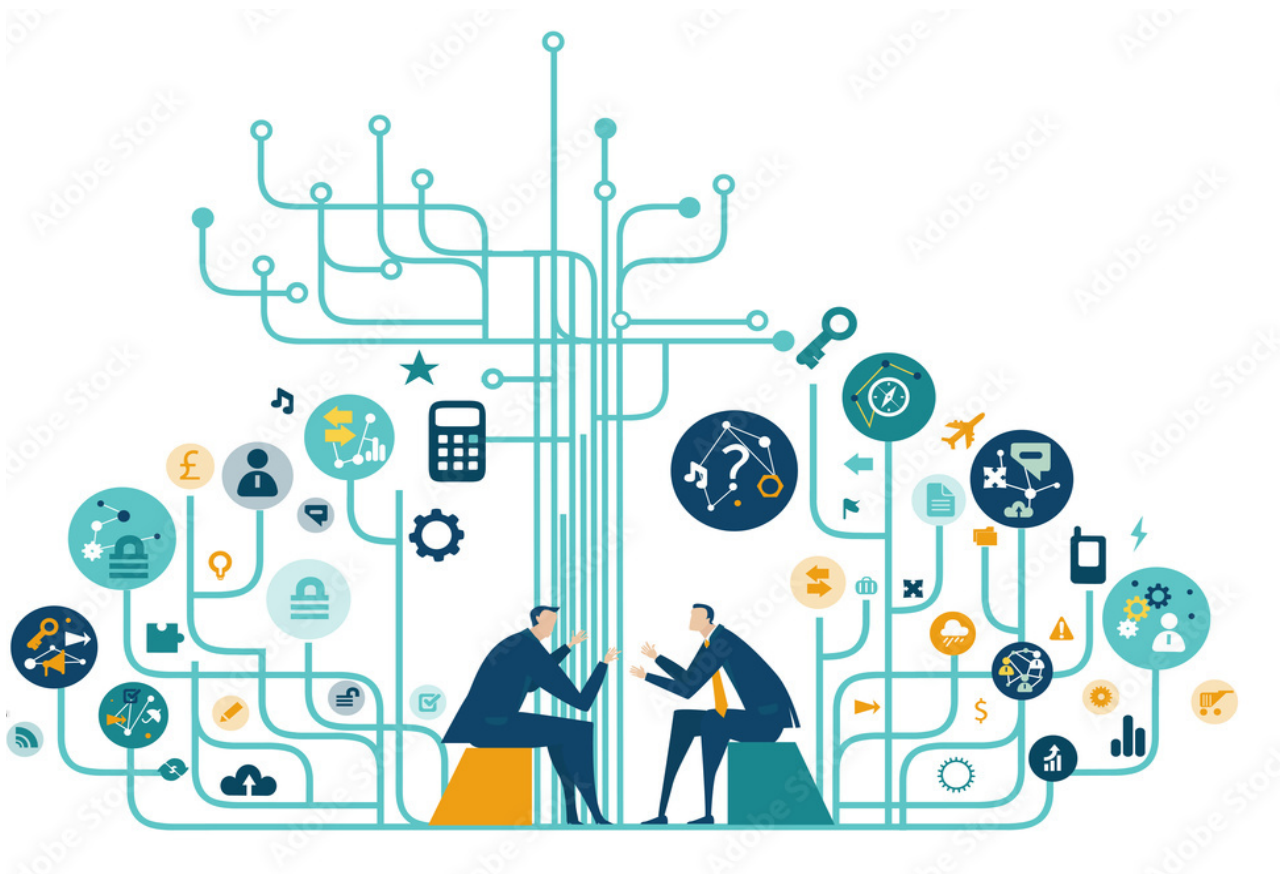
```
R2 score: 0.8301806737108361
MSE: 0.14928243962393878
MAE: 0.296490721288515
```

Tabulation Of Results From Different Algorithms Applied :-

Algorithm	Most Influential Factor	Accuracy
Decision Tree Regressor	Course_Material	0.9556
Random Forest Regressor	Course_Material	0.8301

The **decision tree regressor** has a **higher accuracy of 0.955** compared to the random forest regressor with an accuracy of 0.8301. The decision tree algorithm tends to **overfit** the training data.

While the **random forest algorithm** tries to reduce overfitting by aggregating multiple decision trees. However, in this case, the decision tree algorithm performs better, possibly because the data is not very complex. The random forest algorithm may perform better on more complex datasets.



CONCLUSION FROM THE RESULTS :-

The results from our decision tree and random forest models reveal that the most influential factor on a student's SGPA is the Course Materials that are provided to students for their study. This aligns with the findings of Fook and Sidhu (2015) who investigated learning challenges faced by students in higher education and found that students often struggle with time management and exam preparation. Additionally, Reuter et al. (2017) identified exams as a learning obstacle in higher software engineering education.

Our results also show that the decision tree model outperforms the random forest model in terms of accuracy, with an accuracy score of 0.955 compared to 0.8301 for random forest. The study by Liao and Wu (2022) used multimodal learning analytics models to explore the impact of digital distraction and peer learning on student performance. Although our study focused only on identifying the most influential factor on SGPA, their findings on the impact of digital distraction could be incorporated in future studies to investigate its role as a predictor of SGPA.

Finally, Hershner and Chervin (2014), Phelan et al. (1991), and Carroll (2022) provide valuable insights into other factors that may influence a student's academic performance, such as sleepiness, cultural background, and self-talk. Future studies could consider incorporating these factors in predicting academic performance.